IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Patent Application for:

BI-DIRECTIONAL INDICES FOR TRICK MODE VIDEO-ON-DEMAND

Inventor(s): Leo Mark Pedlow, Jr. and Davender Agnihotri

Docket Number: SNY-T5710.01

Prepared By:

Miller Patent Services
2500 Dockery Lane
Raleigh, NC 27606

Phone: (919) 816-9981
Fax: (919) 816-9982
Email: miller@patent-inventions.com

CERTIFICATE OF EXPRESS MAILING FOR NEW PATENT APPLICATION

"Express Mail" mailing label number: ER 126259245 US

Date of Deposit: 1-23-04

I Hereby certify that this paper or fee is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to the Commissioner for Patents, PO Box 1450, Alexandria, VA 22313-1450.

Typed or printed name of person mailing paper or fee: Catherine N. Miller

Signature of person mailing paper or fee: Catherine N. Miller

BI-DIRECTIONAL INDICES FOR TRICK MODE VIDEO-ON-DEMAND

CROSS REFERENCE TO RELATED DOCUMENTS

This application is related to and claims priority benefit of U.S. Provisional Patent Application Serial No. 60/516,157 filed October 31, 2003 to Pedlow et al. for "Bi-Directional Indices for Trick Mode Navigation of Video On Demand Playback" which is hereby incorporated by reference. This application is also related to U.S. Patent Applications docket number SNY-R4646.01 entitled "Critical Packet Partial Encryption" to Unger et al., serial number 10/038,217; patent applications docket number SNY-R4646.02 entitled "Time Division Partial Encryption" to Candelore et al., serial number 10/038,032; docket number SNY-R4646.03 entitled "Elementary Stream Partial Encryption" to Candelore, serial number 10/037,914; docket number SNY-R4646.04 entitled "Partial Encryption and PID Mapping" to Unger et al., serial number 10/037,499; and docket number SNY-R4646.05 entitled "Decoding and Decrypting of Partially Encrypted Information" to Unger et al., serial number 10/037,498 all of which were filed on January 2, 2002 and are hereby incorporated by reference herein.

COPYRIGHT NOTICE

and Trademark Office patent file or records, but otherwise reserves all copyright rights whatsoever.

BACKGROUND

The Passage™ initiative (Passage is a trademark of Sony Electronics Inc.), promoted by Sony, provides a mechanism for MSOs (Multiple Service Operators) to deploy non-legacy headend equipment, subscriber devices and services on their existing legacy networks. At present, in the USA, these networks are most commonly supplied by either Motorola (formerly General Instrument) or Scientific Atlanta. These two companies at present constitute better than a 99% share of the US cable system market as turnkey system providers. The systems, by design, employ proprietary technology and interfaces precluding the introduction of non-incumbent equipment into the network. An MSO, once choosing one of these suppliers during conversion from an analog cable system to a digital cable system, faces a virtual monopoly when seeking suppliers for additional equipment as their subscriber base or service offering grows.

Before the Passage™ initiative, the only exit from this situation was to forfeit the considerable capital investment already made with the incumbent provider, due to the intentional incompatibility of equipment between the incumbent and other sources. One primary barrier to interoperability is in the area of conditional access (CA) systems, the heart of addressable subscriber management and revenue collection resources in a modern digital cable network.

The Passage™ technologies were developed to allow the independent coexistence of two or more conditional access systems on a single, common plant. Unlike other attempts to address the issue, the two systems operate with a common transport stream without any direct or indirect interaction between the conditional access systems. Some of the basic processes used in these technologies are discussed in detail in the above-referenced pending patent applications.

The above-referenced commonly owned patent applications, and others, describe inventions relating to various aspects of methods generally referred to herein as partial

encryption or selective encryption, consistent with certain aspects of Passage™. More particularly, systems are described therein wherein selected portions of a particular selection of digital content are encrypted using two (or more) encryption techniques while other portions of the content are left unencrypted. By properly selecting the portions to be encrypted, the content can effectively be encrypted for use under multiple decryption systems without the necessity of encryption of the entire selection of content. In some embodiments, only a few percent of data overhead is consumed to effectively encrypt the content using multiple encryption systems. This results in a cable or satellite system being able to utilize Set-top boxes (STB) or other implementations of conditional access (CA) receivers (subscriber terminals) from multiple manufacturers in a single system - thus freeing the cable or satellite company to competitively shop for providers of Set-top boxes.

In each of these disclosures, the clear content is identified using a primary Packet Identifier (PID). A secondary PID (or shadow PID) is also assigned to the program content. Selected portions of the content are encrypted under two (or more) encryption systems and the encrypted content transmitted using both the primary and secondary PIDs (one PID or set of PIDs for each encryption system). The so-called legacy STBs operate in a normal manner decrypting encrypted packets arriving under the primary PID and ignoring secondary PIDs. The newer (non-legacy) STBs operate by associating both the primary and secondary PIDs with a single program. Packets with a primary PID are decoded normally and packets with a secondary PID are first decrypted then decoded. The packets associated with both PIDs are then assembled together to make up a single program stream. The PID values associated with the packets are generally remapped to a single PID value for decoding (e.g., shadow PIDs remapped to the primary PID value or vice versa).

One aspect of VOD that has become a "signature" feature is the support of "trick modes". These are operational modes invoked by the session client that mimic a traditional VCR or DVD player and includes fast forward, rewind, pause, suspend (stop), slow motion, etc. Trick modes have been heretofore implemented through the creation of

multiple files containing a subset of the original content. The techniques used heretofore in actual cable systems have been wasteful of storage space.

BRIEF DESCRIPTION OF THE DRAWINGS

Certain illustrative embodiments illustrating organization and method of operation, together with objects and advantages may be best understood by reference detailed description that follows taken in conjunction with the accompanying drawings in which:

FIGURE 1 is a block diagram of a clear video VOD system.

FIGURE 2 is a diagram illustrating storage of I-frame data to support trick mode operation in a VOD system.

FIGURE 2A is a diagram illustrating Fast Forward Trick mode storage.

FIGURE 2B is a diagram illustrating Fast Reverse Trick mode storage.

FIGURE 3 is a block diagram illustrating a first embodiment of a more storage efficient trick play arrangement consistent with certain embodiments of the present invention.

FIGURE 3A is a diagram illustrating bi-directional trick mode storage consistent with certain embodiments of the present invention.

FIGURE 4 is a flow chart depicting a process of playing content in a system such as that of **FIGURE 3** consistent with certain embodiments of the present invention.

FIGURE 5 is a block diagram illustrating an even more storage efficient trick play arrangement consistent with certain embodiments of the present invention.

FIGURE 6 is a flow chart describing one embodiment of the operation of the system of **FIGURE 5** in a manner consistent with certain embodiments of the present invention.

FIGURE 7 is another flow chart describing another embodiment of the operation of the system of **FIGURE 5** in a manner consistent with certain embodiments of the present invention.

ACRONYMS, ABBREVIATIONS AND DEFINITIONS

ASI - Asynchronous Serial Interface

CA - Conditional Access

CASID - Conditional Access System Identifier

CPE - Customer Premises Equipment

DHEI - Digital Headend Extended Interface

ECM - Entitlement Control Message

EPG - Electronic Program Guide

GOP - Group of Pictures (MPEG)

MPEG - Moving Pictures Experts Group

MSO - Multiple System Operator

PAT - Program Allocation Table

PID - Packet Identifier

PMT - Program Map Table

PSI - Program Specific Information

QAM - Quadrature Amplitude Modulation

RAM - Random Access Memory

SAN - Storage Area Network

VOD - Video on Demand

Critical Packet - A packet or group of packets that, when encrypted, renders a portion of a video image difficult or impossible to view if not properly decrypted, or which renders a portion of audio difficult or impossible to hear if not properly decrypted. The term "critical" should not be interpreted as an absolute term, in that it may be possible to hack an elementary stream to overcome encryption of a "critical packet", but when subjected to normal decoding, the inability to fully or properly decode such a "critical packet" would inhibit normal viewing or listening of the program content.

Selective Encryption (or Partial Encryption) – encryption of only a portion of an elementary stream in order to render the stream difficult or impossible to use (i.e., view or hear).

Dual Selective Encryption – encryption of portions of a single selection of content under two separate encryption systems.

Passage™ - Trademark of Sony Electronics Inc. for various single and multiple selective encryption systems, devices and processes.

Trick mode – an operational mode of playback of digital content to simulate fast forward, fast reverse (rewind), pause, suspend (stop), slow motion, etc. operations as in a video tape system.

The terms "a" or "an", as used herein, are defined as one, or more than one. The term "plurality", as used herein, is defined as two or more than two. The term "another", as used herein, is defined as at least a second or more. The terms "including" and/or "having", as used herein, are defined as comprising (i.e., open language). The term "coupled", as used herein, is defined as connected, although not necessarily directly, and not necessarily mechanically. The term "program", as used herein, is defined as a sequence of instructions designed for execution on a computer system. A "program", or "computer program", may include a subroutine, a function, a procedure, an object method, an object implementation, in an executable application, an applet, a servlet, a source code, an object code, a shared library / dynamic load library and/or other sequence of instructions designed for execution on a computer system.

The terms "scramble" and "encrypt" and variations thereof may be used synonymously herein. Also, the term "television program" and similar terms can be interpreted in the normal conversational sense, as well as a meaning wherein the term means any segment of A/V content that can be displayed on a television set or similar monitor device. The term "storing" as used herein means both the act of placing data into a storage medium and holding the data in storage in the storage medium. The term "video" is often used herein to embrace not only true visual information, but also in the conversational sense (e.g., "video tape recorder") to embrace not only video signals but associated audio and data. The term "legacy" as used herein refers to existing technology used for existing cable and satellite systems. The exemplary embodiments of VOD disclosed herein can be decoded by a television Set-Top Box (STB), but it is

contemplated that such technology will soon be incorporated within television receivers of all types whether housed in a separate enclosure alone or in conjunction with recording and/or playback equipment or Conditional Access (CA) decryption module or within a television set itself.

DETAILED DESCRIPTION

While this invention is susceptible of embodiment in many different forms, there is shown in the drawings and will herein be described in detail specific embodiments, with the understanding that the present disclosure of such embodiments is to be considered as an example of the principles and not intended to limit the invention to the specific embodiments shown and described. In the description below, like reference numerals are used to describe the same, similar or corresponding parts in the several views of the drawings.

A generalized VOD system 10, as shown in **FIGURE 1**, contains some or all of the following elements / resources: Content Aggregation and Asset management 14, Content distribution (SAN) 18, Video server module(s) 22, Session Management 26, Transaction management 30, Billing system 34, EPG server or VOD catalog server 38, Transport router/switch fabric (routing matrix) 42, Stream encryption device(s) (not shown in this Figure), and QAM modulators/upconverters and other edge resources 46. This VOD system 10 provides programming to the subscriber terminals such as 50 for ultimate viewing and listening on a TV set or other monitor device 54.

In operation, content is received from various sources including, but not limited to, satellite broadcasts received via one or more satellite dishes 58. Content is aggregated at 14 and cataloged at EPG server or VOD catalog server 38. Content is then distributed at 18 to one or more video servers 22. When a subscriber orders a VOD selection, a message is sent from the subscriber terminal (e.g., STB) 50 to the session manager 26. The session manager 26 notifies the transaction manager 30 to assure that the billing system 34 is properly brought into play. The session manager 26 selects a VOD server from a cluster of VOD servers having the requested content on it and having a signal path

that reaches the node serving the subscriber. The session manager also enables the routing matrix 42 to properly route the selected video content through the correct edge resources 46 for delivery to the subscriber terminal 50.

For MPEG and similar digital coding schemes, two basic types of frames of video are used: Intra-coded pictures and Inter-coded (or predictive coded) pictures. In MPEG, Intra-coded pictures are commonly referred to as I-pictures or I-frames. Inter-coded pictures in MPEG are further categorized as Predictive coded pictures (P-frames or P-pictures) and Bi-directional coded pictures (B-frames or B-pictures).

As previously mentioned, one aspect of VOD that has become a "signature" feature is the support of so-called "trick modes". These are operational modes invoked by the session client that mimic a traditional VCR or DVD player and includes fast forward, fast reverse (rewind), pause, suspend (stop), slow motion, etc. Fast forward and fast reverse (FF and FR) trick modes have been heretofore implemented through the creation of multiple files containing a subset of the original content (subfiles) as illustrated in **FIGURE 2**. The content is generally stored in a set of RAID (Redundant Array of Independent Disks) drives 70. A particular selection of content is stored in its entirety in a file 74 within the RAID drives 70. A set of subfiles for fast reverse and fast forward trick modes (files 78 and 80 respectively) contain I-frames ordered in a manner that will permit sequential playback to achieve the fast reverse (rewind) and fast forward playback effect. That is, file 78 contains I frames in reverse order while file 80 contains I frames in forward order. Typically, these subfiles contain only I-frames, since I-frames contain stand-alone whole pictures (see ISO/IEC 13818-2, section 6.1.1.7). I-frames are somewhat larger than B or P frames, and they typically represent approximately as much as 21% of the data in a given video selection. For purposes of this document, MPEG I-frames and B- or P-frames will be used as examples without limitation, since similar or equivalent frame structures are used for other compressed video formats.

FIGURE 2A and **2B** further illustrate this arrangement. With reference to **FIGURE 2A**, in order to do Fast Forward trick mode play, the I frames from file 74 are stored in file 80. Index table 88 relates the full content file 74 to FF trick mode file 80 by

indexing the file pointers for the I frames in the full content file 74 with corresponding file pointers in file 80. Playback of the file at normal speed is carried out in the direction of arrow 92. Fast Forward trick mode playback is in the direction of arrow 94.

With reference to **FIGURE 2B**, in order to do Fast Reverse trick mode play, the I frames from file 74 are stored in file 78, in this example, in reverse order. Index table 96 relates the full content file 74 to FR trick mode file 78 by indexing the file pointers for the I frames in the full content file 74 with corresponding file pointers in file 78. Playback of the file at normal speed is carried out in the direction of arrow 92. Fast Reverse trick mode playback is in the direction of arrow 98.

A file containing only I-frames extracted from the original content affords the ability to have accelerated playback, since typical GOP (group of pictures) structures have only one frame in about 10 to 20 as an I-frame. If the I-frame files are played at normal rates (1 frame per 33 mS) the pictures will <u>appear</u> to the viewer to sequence at about a 10x to 20x rate, though the actual data rate is the same as the original content. If the I-frame sequence is reversed in the file, the motion will appear to run backwards. This is the method used to implement fast forward and fast-reverse (rewind) trick modes.

By attaching an index count to match the I-frames in the original content file to the duplicated I-frames stored in the associated subfiles 78 and 80 (for FF or FR, respectively), a method is provided to allow immediate transition from normal speed forward play to fast forward or fast reverse (rewind) play. In operation the video server plays the selected content file and upon subscriber selection of a trick mode (or vice versa) the server notes the index value of the closest I-frame and then opens the appropriate associated subfile 78 or 80 and moves to the I-frame in the subfile with the same corresponding index. The video server treats all stream content (main file or subfiles) the same and always spools the MPEG packets to the outgoing transport stream at the same constant bit rate through multiplexers and buffers 84 as shown. It is through this method that trick modes are typically implemented on a slotted, session based system without the encumbrance of additional, dynamic bit rate issues.

A function of the VOD video server(s) 22, in addition to origination of session A/V content, is the creation of the associated, session specific PSI (program specific information). This information is a departure from the broadcast model in that the PSI is extremely dynamic. The content of the PAT and subordinate PMTs change whenever a new session is started or ended. In the broadcast world, the PSI changes very seldom because the PSI tables reflect only the structure of the transport multiplex, not the actual A/V content carried within.

The VOD video server 22 dynamically assigns a new session to an existing, available "slot" in an outgoing transport multiplexed stream. The slot is denoted by the MPEG program number and in many cases, the combination of which transport stream (TSID) and program number determine at the service level a unique session and the routing that occurs as a result. Edge resources 46 generally are not configured dynamically. The routing of content appearing on a particular input port to a specific QAM carrier at the output is determined through a preconfigured, static assignment of TSID/input port and program number mapping to specific QAM resources in the device. This same mapping information is also loaded in the VOD system so that once a session is requested by and authorized for a specific subscriber terminal 50, a solution to a routing matrix 42 can be determined to find the appropriate VOD server 22 and QAM transport 46 serving the requestor. This solution also considers dynamic issues such as which servers 22 the requested asset is loaded upon, and server loading/available slots in addition to the simpler, static solution to finding the first possible path to the requesting subscriber terminal 50.

In addition to solving the routing matrix 42 and provisioning the session with PIDs and PSI appropriate to follow the intended route, elements of the same information (program ID and QAM frequency) are also communicated to the session client at subscriber terminal 50 at the subscriber's premises so that the requested stream can be properly received and presented to the subscriber.

Perhaps the simplest VOD distribution system implementation is a clear VOD distribution system, i.e. one that contains no encryption as depicted in **FIGURE 1**. The

system of **FIGURE 1** is used as an example system, but embodiments of the inventions disclosed herein can also be used with other VOD distribution systems. While not providing any safekeeping of what might be considered the entertainment medium's most valuable properties, namely current feature films, etc., clear VOD avoids many of the issues that the incumbent cable system providers to date have not adequately addressed and that introduction of a second, alternative CA system complicates even further still. Throughout this discussion, it is instructive to carry an example VOD movie through the various embodiments to illustrate the relative storage efficiencies obtained with the various systems disclosed. A real world example of a VOD movie, which will be used throughout this document, has the following attributes:

Compressed video data rate:	3Mbit/S
Movie length:	120 minutes (2 Hrs)
I-frame overhead:	17%
Total storage used for the video portion of a single, clear (unencrypted) copy of a film:	**3.618GBytes.**

To reduce storage requirements for each movie and to improve system performance, the VOD file architecture shown in **FIGURE 2** can be modified to remove the second "trick" mode file containing reversed sequence video (used for fast rewind display), as will be described in connection with **FIGURE 3**. Instead, a second set of indices is created pointing to the single, remaining "trick" mode file and containing the same sequence data, but having the indices in reversed sequence for reversed motion. This allows forward or reverse navigation of the single "trick" mode file by simply choosing the appropriate set of playback indices indicating the correct sequence of frames to queue into the playback buffer for output to the client terminal for display. Equivalently, the same set of indices can be used for FF or FR, with the indices sequentially traversed either in the forward or reverse direction respectively.

Referring to **FIGURE 3**, content retrieved from RAID mass storage files 70 is stored at 200 (either encrypted or in the clear). A single set of trick mode content (e.g., I frames in an MPEG encoded selection of content) is stored in a separate file 220. This content may represent as much as approximately 21% of the original file. Thus, by elimination of two such files, by use of forward reference indices 222 and reverse reference indices 224 (or equivalently, a single index table that is traversed either forward or backward), up to approximately 21% of storage space can be saved. In this case, the forward and reverse indices are the same except for their order.

The indices can be visualized as a table such as **TABLEs 1** and **2** below. In the case of the using a forward and a reverse index, **TABLE 1** represents the forward index and **TABLE 2** represents the reverse index.

TABLE 1 (222)	
File pointers in Normal Play File arranged in descending order (forward time order)	File pointers in Trick Play File arranged in descending order (forward time order)

TABLE 2 (224)	
File pointers in Normal Play File arranged in descending order (forward time order)	File pointers in Trick Play File arranged in ascending order (reverse time order)

When a subscriber is playing a file in a normal playback mode, data are spooled sequentially from the Normal Play File 200. When a trick play mode of fast forward is initiated, a location in the Trick Mode File 220 is identified by finding the closest file pointer corresponding to the current file pointer in the Normal Play File 200 by reference to **TABLE 1**. Data are then spooled from the trick play file in the order dictated by the file pointers in **TABLE 1**.

In a similar manner, when a subscriber is playing a file in a normal playback mode, data are spooled sequentially from the Normal Play File 200. When a trick play mode of fast reverse is initiated, a location in the Trick Mode File 220 is identified by finding the closest file pointer corresponding to the current file pointer in the Normal Play File 200 by reference to **TABLE 2**. Data are then spooled from the trick play file in the order dictated by the file pointers in **TABLE 2**.

In either case, when the subscriber returns to normal play mode, the current file index in the Trick Mode File 220 is referenced to a file pointer in the Normal Play File 200 using the appropriate table in order to return to the approximately same location for normal playback. It is noted that slight differences in the locations between the Normal Play file index and the Trick Mode indices will occur since there is not a one-to-one correspondence between the pointers, but the loss of continuity is on the order of a fraction of a second and is generally not noticeable to the viewer.

As noted earlier, a similar result can be achieved with a single set of file indices such as that shown in **TABLE 3** (The trick play file pointers could be either ascending or descending.). In this example, fast forward trick play is achieved by playing out the trick play file 220 in the forward direction of the file pointers (top to bottom), and fast reverse trick play is achieved by playing out the trick play file 220 in the reverse direction of the file pointers (bottom to top).

FF ↓ FR ↑

TABLE 3	
File pointers in Normal Play File arranged in descending order	File pointers in Trick Play File arranged in descending order

With reference to **FIGURE 3A**, in order to do Fast Forward or Fast Reverse trick mode play, the I frames from file 200 are stored in file 220. Index table 222 relates the full content file 200 to trick mode file 220 by indexing the file pointers for the I frames in

the full content file 74 with corresponding file pointers in file 220. Additionally, in this exemplary embodiment, the frame length is indexed in order provide the information needed to jump from frame to frame in Fast Reverse trick play mode. Playback of the file at normal speed is carried by playing out the content of file 200 out in the direction of arrow 200. Fast Forward trick mode playback is carried out by playback of file 220 in the direction of arrow 224. In Fast Reverse trick mode, playback is carried out in the directions indicated by arrows 228. Thus, I frame I_{n+3} is played out starting at index 0x7D7 for a duration of 0x275, followed by I frame I_{n+2} played out starting at index 0x408 for a duration of 0x3CF, etc.

Thus, for a single trick mode file and index table Fast Forward trick mode is a linear action in terms of play out of the content of file 220. The process simply jumps to the correlated point in the trick mode file and begins playing out the disk content by monotonically advancing the file pointer. This is not the case for Fast Reverse trick mode. In FR trick mode, the frames are played based upon spolling the data for a single frame by advancing the file pointer until it gets to a terminal count for the frame. Then the file pointer is reverse jumped back to the preceding frame and frame content is linearly spooled by advancing the file pointer again until the end of the frame as indicated by the frame length entry in table 222.

One process consistent with certain embodiments is depicted as process 230 of **FIGURE 4** starting at 232. At 234, the full content is stored in a first file 200 (the main content file used for normal playback). The trick play content is stored in a second file at 236. At 238, an index is created such as that depicted in **TABLE 3** that relates the first and second files to one another. When a subscriber issues a valid request for playback at 242, if no trick mode is invoked at 244, the content is retrieved from the first file in forward order at 248 and the content is spooled to an output stream at 252. When the end of the file is reached at 256, the process stops at 258. Otherwise, the process periodically returns to 244 (e.g., using interrupts or any other suitable program mechanism) to determine if a trick play mode has been invoked by the subscriber.

If a trick play mode is invoked by the subscriber at 244, the index table is referenced to identify a frame of video in the second file (trick play file 220) that is near the current point being played back (for example, the next intra-coded frame in the second file). If, at 266, the trick play mode selected is the Fast Forward (FF) mode, a frame (or group of frames) is retrieved from the second file advancing in the direction representing forward time at 270. This content is then spooled to the output at 278 and if the end of the content is not reached at 282, and the mode (normal, FF or FR) mode of the playback has not changed at 284, control returns to 270 where more data are retrieved, again progressing in the forward time direction through the second file. When the end is reached at 282, the process ends at 258.

If the trick play mode invoked at 266 is fast reverse (FR or Rewind), control passes from 266 to 274 where a frame (or group of frames) is retrieved from the second file advancing in the direction representing reverse time. Control then passes to 278.

If, at 284, the subscriber changes the mode of playback back to normal speed playback, control passes to 290 where the index table is again referenced to identify the location in the first file corresponding the current point of playback from the second file. The first file is then entered at this point and control passes to 248 where normal speed playback proceeds as previously described. Also, if the subscriber changes the mode from FF to FR at 284, control passes to 274, and if the subscriber changes the mode of playback from FR to FF at 284, control passes to 270.

Many variations in this process are possible without departing from certain embodiments consistent with the present invention. For example, the ordering of certain actions can be rearranged without changing the basic operation. Also, equivalently, two tables such as **TABLE 1** and **TABLE 2** could be used. In this equivalent example, instead of designating an order of retrieval from the second file at 270 and 274, the order is always in the same direction, but with reference to a different table. Also in this variation, the tables used to determine entry points in the files at 262 and 290 depend upon the trick mode selected, thus a mode determination is made at 262 to determine

which table to use. Other variations will also occur to those skilled in the art upon consideration of the present teaching.

Thus, a method of storing digital video content to facilitate trick play, the content comprising intra-coded frames of video and inter-coded frames of video, consistent with certain embodiments, involves: storing the inter-coded and the intra-coded frames of the content in a first file; storing a duplicate of the inter-coded frames of the content in a second file; storing a set of forward indices that relates the intra coded frames with the inter-coded frames in a forward direction such that playback of the second file in the order of the forward indices simulates a fast-forward playback; and storing a set of reverse indices that relates the intra-coded frames with the inter-coded frames in a reverse direction such that playback of the second file in the order of the reverse indices simulates a fast-reverse playback.

Another method of storing digital video content to facilitate trick play, the content comprising intra-coded frames of video and inter-coded frames of video, consistent with certain embodiments involves: storing the inter-coded and the intra-coded frames of the content in a first file; storing the intra-coded frames of the content in a second file; storing a set of indices that relate the intra-coded frames in the first file with the intra-coded frames in the second file, such that playback of the second file simulates a fast-forward playback if played back in a first order and simulates a fast rewind if played back in a second order.

It is noted that although the arrangement of **FIGURE 3** provides substantial savings in storage space over the techniques currently in use, additional savings in storage space can be realized by the recognition that the information stored in the trick mode content file 220 is redundant to the I frames stored in the normal play content file 200. By spooling normal play content from both files, an additional savings of up to approximately 21% can be realized as depicted in **FIGURE 5**. In this illustration, all I-frame data (inter-coded data) are stored in the trick mode content file 320, and supplemental normal play content (intra-coded data, B and P frame data) is stored in the normal play content file 300. The bidirectional indices concept is extended for even

further storage economy in this embodiment. If one recognizes that the normal mode playback file contains a duplication of the same I-frames played in "trick" modes, a dynamic architecture can be created to remove any redundant I-frame content from the normal mode playback file. During normal playback, the two files are "blended" (normal play and "trick" modes), while only the I-frame sequences in the "trick" mode file are accessed during fast forward, fast reverse (rewind), etc.

As noted above, if one takes the concept described above in connection with **FIGURE** 3 one step further, the current convention in VOD systems to store the same I-frames of a movie in forward and reversed sequence to allow fast forward and rewind "trick" modes can be eliminated. An illustration of this concept is shown in the example of **FIGURE 5**. These dual files for forward and reverse are replaced by a single file 320 of I-frames in normal forward sequence with two sets of indices, one set 322 for playing the I-frame file in forward order and one set 324 for playing the I-frame file in reverse order, or equivalently, by a single index that is traversed in the forward or reverse direction for FF or FR play respectively. The appropriate sets of indices are chosen depending on whether forward or reverse high-speed motion is desired. The forward indices are also used to reconstruct the normal speed stream when matching the I-frame file to the non-critical content file to reconstruct the entire stream. On a clear or re-encrypted VOD system, this will allow up to about 21% storage savings. On a composite pre-encrypted storage system, up to about 42% storage savings may be realized

If this additional opportunity is taken, a significant storage economy can be realized over <u>all</u> VOD schemes, including traditional (unencrypted) VOD, as deployed today. The traditional VOD video server has three files for each feature or movie: two containing just I-frames (one in reverse order) and one containing the complete original copy. Research on encoded streams conducted by Sony has shown that the I-frames generally represent approximately 12 to 21% of the total content, typically about 17%. Thus, by using bidirectional indices and dynamic composition, this method removes the redundant I-frames from the clear stream file and an additional (nominal) 17% storage savings is realized over using bidirectional indices alone. This indicates a potential 34%

(nominal, 42% maximum) video server disk storage savings for an existing VOD system described in **FIGURE 2**.

As with the example of **FIGURE 3-3A**, either two reference tables or one could be used in implementing various embodiments consistent with this example. In this example, however, it should be remembered that the normal play file does not contain a full set of content, but rather may contain only data associated with intra-coded frames. Thus, to carry out a normal playback, the index tables are used to identify a full set of data and data are pulled from both file 300 and file 320.

The indices can be visualized as a table such as **TABLEs 4** and **5** below. In the case of the using a forward and a reverse index, **TABLE 4** represents the forward index and **TABLE 5** represents the reverse index.

TABLE 4 (322)	
File pointers in Normal Play File arranged in descending order File pointers point to intra-coded data	File pointers in Trick Play File arranged in descending order File pointers point to inter-coded data

TABLE 5 (324)	
File pointers in Normal Play File arranged in descending order File pointers point to intra-coded data	File pointers in Trick Play File arranged in ascending order File pointers point to inter-coded data

When a subscriber is playing a file in a normal playback mode, data are spooled sequentially by alternating retrieval of data from the Normal Play File 300 and the Trick Mode File 320. When a trick play mode of fast forward is initiated, a location in the Trick Mode File 320 is identified by finding the closest file pointer corresponding to the

current file pointer by reference to **TABLE 4**. Data are then spooled only from the trick play file in the order dictated by the file pointers in **TABLE 4**.

In a similar manner, when a subscriber is playing a file in a normal playback mode, data are spooled sequentially from both the Normal Play File 300 and the Trick Mode File 320. When a trick play mode of fast reverse is initiated, a location in the Trick Mode File 320 is identified by finding the closest file pointer to the current playback point by reference to **TABLE 5**. Data are then spooled from the trick play file in the order dictated by the file pointers in **TABLE 5**.

In either case, when the subscriber returns to normal play mode, the current file index in the Trick Mode File 320 is used as a starting location for normal play. Data are then pulled from both files 300 and 320 to produce normal playback. It is noted that there is no overlap in the locations between the Normal Play file index and the Trick Mode indices. Playback will generally alternate between playing one or more I frames from file 320 and playing one or more B or P frames from file 300 to construct a complete set of the content.

As noted earlier, a similar result can be achieved with a single set of file indices such as that shown in **TABLE 6** (The trick play file pointers could be either ascending or descending.). In this example, fast forward trick play is achieved by playing out the trick play file 320 in the forward direction of the file pointers (top to bottom), and fast reverse trick play is achieved by playing out the trick play file 320 in the reverse direction of the file pointers (bottom to top). Again, normal playback involves selecting data from both files.

FF ↓ FR ↑

TABLE 3	
File pointers in Normal Play File arranged in descending order File pointers point to intra-coded data	File pointers in Trick Play File arranged in descending order File pointers point to inter-coded data

A process 330 for playback of content using the arrangement depicted in **FIGURE 5** is shown in **FIGURE 6** starting at 332. At 334, intra-coded frames are stored in a first file (300). At 336, inter-coded frames are stored in a second file (320). At 338 one or more index tables are created and stored that relate the first file to the second file. In this example, a single index table is depicted. When a subscriber initiates a playback at 342, a determination of playback mode is made at 344. If a normal playback mode has been invoked at 344, intra-coded frames from the first file and inter-coded frames from the second file are retrieved at 348 and assembled in forward sequence at 352 to produce full motion content. This content is then spooled to the output at 356 until the end is reached at 358 in which case the process stops at 360. If the end is not reached, control returns to 344 on a periodic or frequent basis to determine if a trick mode has been invoked by the subscriber.

If a trick mode has been invoked at 344, a location in the second file is identified, by reference to **TABLE 6**, that is close to the current point of playback (e.g., the next inter-coded frame) at 364. If a fast forward trick mode has been invoked at 368, control passes to 372 where inter-coded frames are retrieved in forward order from the second file. If fast reverse trick mode has been invoked, control passes from 368 to 380 where inter-coded frames are retrieved in reverse order from the second file. In either case, the retrieved frames are spooled to the output at 376. If the end of the file is reached at 388, the process stops at 360. Otherwise, control passes back to 344 to monitor the state of the selection of trick mode and either continue to operate in trick mode, change from one trick mode to the other or return to normal playback mode.

Many variations in this process are possible without departing from certain embodiments consistent with the present invention. For example, the ordering of certain actions can be rearranged without changing the basic operation. Also, equivalently, two tables such as **TABLE 4** and **TABLE 5** could be used. In this equivalent example, instead of designating an order of retrieval from the second file at 372 and 380, the order

is always in the same direction, but with reference to a different table. Also in this variation, the tables used to determine entry points in the files at 364 and for normal playback depends upon the trick mode selected, thus a mode determination is made at 364 to determine which table to use. Other variations will also occur to those skilled in the art upon consideration of the present teaching.

FIGURE 7 shows another embodiment of a playback process similar to that of **FIGURE 6**, but detailing certain variations starting at 402. In this embodiment, processes 334, 336, 338, 342, 344, 364 and 368 can be similar to corresponding processes in process 330. Also, to simplify the diagram, the end of file operation has been omitted, but adding it will be clear to those skilled in the art upon consideration of the present teaching.

In the normal play mode decision from 344, a determination is made as to whether or not the first (or next) frame for playback is located in the second file. If so, the next frame is retrieved from the second file at 408. If not, the next frame is retrieved from the first file at 410. In either event, the retrieved frame is spooled to the output at 412 and control returns to 344 to determine if a mode change has taken place. In other words, the presence or absence of an entry in the second file that corresponds to a next frame in the content is used to determine if content is retrieved from the first file at 410 or the second file at 408.

When a fast forward trick mode is invoked at 368, inter-coded frames are retrieved from the second file in forward order at 420 and the frame is spooled to the output at 424. If no mode change occurs at 428, the process returns to 420 to retrieve the next frame. If the mode changes to normal playback mode at 428, control returns to 344.

If a fast reverse trick mode is invoked at 368, inter-coded frames are retrieved from the second file in reverse order at 440 and the frame is spooled to the output at 444. If no mode change occurs at 448, the process returns to 440 to retrieve the next frame. If the mode changes to normal playback mode at 448, control returns to 344.

If the mode changes to fast reverse at 428, control is passed to 440. If the mode changes to fast forward at 448, control passes to 420.

Again, many variations in this process are possible without departing from certain embodiments consistent with the present invention. For example, the ordering of certain actions can be rearranged without changing the basic operation, and end of file provisions should be provided. Also, equivalently, two tables such as **TABLE 4** and **TABLE 5** could be used. In this equivalent example, instead of designating an order of retrieval from the second file at 420 and 440, the order is always in the same direction, but with reference to a different table. Also in this variation, the tables used to determine entry points in the files at 364 and for normal playback depends upon the trick mode selected, thus a mode determination is made at 364 to determine which table to use. Other variations including error trapping as well as other considerations will also occur to those skilled in the art upon consideration of the present teaching.

Thus, a method of storing digital video content to facilitate trick play, the content comprising intra-coded frames of video and inter-coded frames of video, consistent with certain embodiments, involves: storing the inter-coded frames of the content in a first file; storing the intra-coded frames of the content in a second file; storing a set of forward indices that relate the intra-coded frames to the inter-coded frames in a forward direction such that playback of the second file in the order of the forward indices simulates a fast-forward playback; and storing a set of reverse indices that relate the intra-coded frames to the inter-coded frames in a reverse direction such that playback of the second file in the order of the reverse indices simulates a fast-reverse playback.

Another method of storing digital video content to facilitate trick play, the content comprising intra-coded frames of video and inter-coded frames of video, consistent with certain embodiments, involves: storing the inter-coded frames of the content in a first file; storing the intra-coded frames of the content in a second file; storing a set of indices that relate the intra-coded frames in the first file with the intra-coded frames in the second file, such that playback of the second file simulates a fast-forward playback if played back in a first order and simulates a fast rewind if played back in a second order.

A video method, consistent with certain embodiments involves retrieving inter-coded video from a first file; retrieving intra-coded video from a second file; and

assembling the inter-coded video with the intra-coded video to produce an assembled selection of video content.

If one refers to the example movie scenario described previously, the same movie using 3.618GB of storage in the clear VOD state would require only 2.700GBytes to store the same content, still supporting "trick" modes, using bidirectional indices.

The present concepts can also be extended advantageously to a selectively encrypted system. If the "trick" mode subfile and the "critical" data encrypted content can be the same, the content is selectively encrypted at approximately a nominal 17% level (approximately 12% to 21%), much higher than the commonly proposed Passage™ encryption level of approximately 2%, but carrying no inherent storage or system capacity costs, as do other schemes. For this system to work, some changes to the video server software design might be necessary, but these changes would be modifications to the existing processes and would not require substantial new development on the part of the server vendor.

A preprocessor can be used to perform selective encryption of content to be loaded onto the VOD video server 22. A modified file protocol can be used to allow the VOD video server 22 to import and associate these files. Either the preprocessor or the VOD video server 22 can be used to perform the indexing. An alternate instantiation can be used to perform all selective encryption pre-processing within the video server itself. This can be accomplished by modifying the video server application to add a pre-processor task as a separate executable, called by the server during the process to prepare content for pre-encryption.

In accordance with certain embodiments wherein the content is selectively encrypted, "critical packets" are selected for encryption according to a suitable selection criterion associated with the selective encryption process such that they encompass all of the I frames. Thus, the content that is stored as shown in **FIGUREs 3** and **FIGURE 5** in file 220 or 320 has "critical" packets while content stored in file 300 of **FIGURE 5** has "non-critical" packets. Content stored in file 200 of **FIGURE 3** is a mixture of critical and non-critical packets. The "critical" packets selected according to the above-

referenced patent applications constitute approximately 2% to 10% of the program (depending upon program content and the selection criteria used to select packets for encryption). A separate copy of the critical content can be maintained for each conditional access system supported by the MSO.

With reference to **FIGURE 5**, when a subscriber session is initiated, the main file 200 containing the normal play content is queued in the video server for playout. In addition, the file containing the trick play packets 220 is also queued for playout. When the program playback is started, the video server reconstructs a single program multiplex in its stream buffer feeding the outgoing transport the correct sequence of packets based upon the indices in the two component files. Although, in general, only about 2-10% of the packets are encrypted in a selective encryption system according to the above pending patent applications, even further security is provided by encryption of all of the I frames in the present embodiment.

The stream files containing "critical" packets may be the same one as the extracted subfile containing all I-frames for "trick" modes, as was described previously. If this opportunity is taken, then a substantial storage economy can be realized over a traditional VOD video server having three files for each feature or movie: two containing just I-frames (one in reverse order) and one containing the complete original copy.

In the example of **FIGURE 5**, the dual files for forward and reverse are replaced by a single file 320 of I-frames in normal forward sequence with two sets of indices, one set 322 for playing the I-frame file in forward order and one set 324 for playing the I-frame file in reverse order. The appropriate sets of indices are chosen depending on whether forward or reverse high-speed motion is desired. The forward indices are also used to reconstruct the normal speed stream when matching the I-frame file to the non-critical content file to reconstruct the entire stream. On a clear or re-encrypted VOD system, this will allow up to about 21% storage savings. On a composite pre-encrypted storage system, up to about 42% storage savings may be realized

In accordance with certain embodiments consistent with the present invention, certain of the functional blocks used to implement the VOD system can be implemented

using a programmed processor such as a general purpose computer. One example of such a functional block is the session manager 26. However, the invention is not limited to such exemplary embodiments, since other embodiments could be implemented using hardware component equivalents such as special purpose hardware and/or dedicated processors. Similarly, general purpose computers, microprocessor based computers, micro-controllers, optical computers, analog computers, dedicated processors, application specific circuits and/or dedicated hard wired logic may be used to construct alternative equivalent embodiments.

Certain embodiments described herein, are or may be implemented using a programmed processor executing programming instructions that are broadly described above in flow chart form that can be stored on any suitable electronic or computer readable storage medium and / or can be transmitted over any suitable electronic communication medium. However, those skilled in the art will appreciate, upon consideration of the present teaching, that the processes described above can be implemented in any number of variations and in many suitable programming languages without departing from embodiments of the present invention. For example, the order of certain operations carried out can often be varied, additional operations can be added or operations can be deleted without departing from certain embodiments of the invention. Error trapping can be added and/or enhanced and variations can be made in user interface and information presentation without departing from certain embodiments of the present invention. Such variations are contemplated and considered equivalent.

Those skilled in the art will appreciate, upon consideration of the above teachings, that the program operations and processes and associated data used to implement certain of the embodiments described above can be implemented using disc storage as well as other forms of storage such as for example Read Only Memory (ROM) devices, Random Access Memory (RAM) devices, network memory devices, optical storage elements, magnetic storage elements, magneto-optical storage elements, flash memory, core memory and/or other equivalent volatile and non-volatile storage technologies without

departing from certain embodiments of the present invention. Such alternative storage devices should be considered equivalents.

Thus, in one example, a computer readable storage device for storage and retrieval of digital video content, consistent with certain embodiments has a computer readable storage device. A first file resides on the storage device for storing inter-coded frames of the digital video content. A second file resides on the storage device for storing intra-coded frames of the digital video content. An index table is also stored on the storage device that relates the intra-coded frames in the first file with the intra-coded frames in the second file, such that playback of the second file simulates a fast-forward playback if played back in a first order and simulates a fast rewind if played back in a second order.

In another example, a computer readable storage device for storage and retrieval of digital video content, consistent with certain embodiments, has a computer readable storage device. A first file resides on the storage device storing inter-coded frames of the digital video content. A second file residing on the storage device storing intra-coded frames of the digital video content in a second file. A forward index table residing on the storage device that relates the intra-coded frames to the inter-coded frames in a forward direction such that playback of the second file in the order of the forward indices simulates a fast-forward playback. A reverse index table residing on the storage device that relates the intra-coded frames to the inter-coded frames in a reverse direction such that playback of the second file in the order of the reverse indices simulates a fast-reverse playback.

While certain illustrative embodiments have been described, it is evident that many alternatives, modifications, permutations and variations will become apparent to those skilled in the art in light of the foregoing description.

What is claimed is: